

Modelling fixation locations using spatial point processes

Simon Barthelmé
Psychology, University of Geneva

Hans Trukenbrod, Ralf Engbert
Psychology, University of Potsdam

Felix Wichmann
Computer Science, University of Tübingen

11th July 2012

Abstract

Whenever eye movements are measured, a central part of the analysis has to do with *where* subjects fixate, and *why* they fixated where they fixated. To a first approximation, a set of fixations can be viewed as a set of points in space: this implies that fixations are spatial data and that the analysis of fixation locations can be beneficially thought of as a spatial statistics problem. We argue that thinking of fixation locations as arising from *point processes* is a very fruitful framework for eye movement data, helping turn qualitative questions into quantitative ones.

We provide a tutorial introduction to some of the main ideas of the field of spatial statistics, focusing especially on spatial Poisson processes. We show how these concepts can be used to clarify many issues surrounding the role of saliency in free-viewing tasks: what saliency models can successfully predict and what they cannot (and why). We have implemented and made available a software package that attempts to make spatial analysis as user-friendly as possible.

Eye movement recordings are some of the most complex data available to behavioural scientists. At the most basic level they are long sequences of measured eye positions, a very high dimensional signal containing saccades, fixations, micro-saccades, drift, and their myriad variations (Ciuffreda and Tannen, 1995). There are already many methods that process the raw data and turn it into a more manageable format, checking for calibration, distinguishing saccades from other eye movements (e.g., Engbert and Mergenthaler, 2006; Mergenthaler and Engbert, 2010), etc. We will here take the liberty of assuming that this important part of the work has already been done, and that the researcher's main point of interest is what subjects looked at during the experiment. In the kind of experiment that will serve as an example throughout the paper, subjects were shown a number of pictures on a computer screen, under no particular instructions. The resulting data are a number of points in space, representing what people looked at in the picture—the fixation locations. The fact that fixations tend to cluster shows that people favour certain locations and not simply explore at random. The natural question to ask then is, why certain locations are preferred and others are not.

We argue here that a very fruitful approach to the problem is to be found in the methods of spatial statistics (Diggle, 2002; Illian et al., 2008). A sizeable part of spatial statistics is concerned with how things are distributed in space, and fixations are things distributed in space. We will introduce the concepts of point processes and latent fields, and explain how these can be applied to fixations. We will show how this lets us put the important (and much researched) issue of low-level saliency on firmer statistical ground. We will begin with simple models and gradually build up to more sophisticated models that attempt to separate the various factors that influence the location of fixations.

In the eye movement literature, fixations are analysed in a myriad different ways, with varying regard for statistical validity. Our contribution is a framework made up of just a few elements (point processes, nonparametric estimation), but, to paraphrase Claude Lévi-Strauss, these are “good to think with”. The additional clarity and sharpness these elements bring is well worth the time one needs to become conversant with them. To make this process as smooth as possible, we have tried to keep the sometimes tricky issues of implementation out of the main text, and all details are given in the appendix.

1 Point processes

We begin with a general overview on the art and science of generating random sets of points in space. It is important to emphasise at this stage that the models we will describe are entirely *statistical* in nature and not mechanistic: they do not assume anything about how saccadic eye movements are generated by the brain (Sparks, 2002). In this sense they are more akin to familiar linear regression models than, e.g., biologically-inspired models of overt attention during reading or visual search (Engbert et al., 2005; Zelinsky, 2008). The goal of our modelling is to provide statistically

sound and useful summaries and visualizations of data, rather than come up with a full story of how the brain goes about choosing where to allocate the next saccade. What we lose in depth, we gain in generality, however: the concepts that are highlighted here are applicable to the vast majority of experiments in which fixations locations are of interest.

1.1 Definition and examples

In statistics point patterns in space are usually described in terms of point processes, which represent realisations from probability distributions over sets of points. Just like linear regression models, point processes have a deterministic and a stochastic component. In linear models, the deterministic component describes the average value of the independent variable as a function of the dependent ones, and the stochastic component captures the fact that the model cannot predict perfectly the value of the independent variable, for example because of measurement noise. In the same way, point processes will have a latent *intensity function*, which describes the expected number of points that will be found in a certain area, and a stochastic part which captures prediction error and/or intrinsic variability.

We focus on a certain class of point process models known as inhomogeneous Poisson processes. Some specific examples of inhomogeneous Poisson processes should be familiar to most readers. These are temporal rather than spatial, which means they generate random point sets in time rather than in space, but equivalent concepts apply in both cases.

In neuroscience, Poisson processes are often used to characterize neuronal spike trains (see e.g., Dayan and Abbott, 2001). The assumption is that the number of spikes produced by a neuron in a given time interval follows a Poisson distribution: for example, repeated presentation of the same visual stimulus will produce a variable number of spikes, but the variability will be well captured by a Poisson distribution. Different stimuli will naturally produce different average spike rates, but spike rate will also vary over *time* during the course of a presentation, for example rising fast at stimulation onset and then decaying. A useful description, summarized in figure 1, is in terms of a latent intensity function $\lambda(t)$ governing the expected number of spikes observed in a certain time window. Formally, $\int_{\tau}^{\tau+\delta} \lambda(t) dt$ gives the expected number of spikes between times τ and $\tau + \delta$. If we note $\mathbf{t} = t_1, t_2, \dots, t_k$ the times at which spikes occurred on a given trial, then \mathbf{t} follows a inhomogeneous Poisson Process (from now on IPP) distribution if, for all intervals $(\tau, \tau + \delta)$:

$$\# \{t_i \in (\tau, \tau + \delta)\} \sim Poi \left(\int_{\tau}^{\tau+\delta} \lambda(t) dt \right) \quad (1)$$

This is statistical shorthand for the “number of spikes occurring in the interval follows a Poisson distribution with mean $\int_{\tau}^{\tau+\delta} \lambda(t) dt$ ”.

The temporal IPP therefore gives us a distribution over sets of points in time (in Figure 1, over the interval $[0, 1]$). Extending to the spatial case is straightforward: we simply define a new intensity function $\lambda(x, y)$ over space, and the IPP now generates point sets such that the expected number of points to appear in a certain area A is $\int_A \lambda(x, y) dx dy$, with the actual quantity again following a Poisson distribution. The spatial IPP is illustrated on Figure 2.

1.2 The whole point of point processes

Given a point set, the most natural question to ask is, generally, “what latent intensity function could have generated the observed pattern?” Indeed, we argue that a lot of very specific research questions are actually special cases of this general problem.

For mathematical convenience, we will from now on focus on the log-intensity function $\eta(x, y) = \log \lambda(x, y)$. The reason this is more convenient is that $\lambda(x, y)$ cannot be negative (we cannot be expecting a negative number of points). $\eta(x, y)$, on the other hand, can take any value whatever, from minus to plus infinity.

At this point we need to introduce the notion of *covariate*, which will be familiar to readers versed in linear models. In statistical parlance, the *response* is what we are interested in predicting, and *covariates* is what we use to predict the response with. In the case of point processes covariates are often spatial functions too.

One of the classical questions in the study of overt attention is the role of low-level cues in attracting gaze. Among low-level cues, local contrast may play a prominent role, and it is a classical finding that observers tend to be more interested in high-contrast regions when viewing natural images (Rajashekar et al., 2007).

Imagine now that our point set $\mathbf{F} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ represents observed fixation locations on a certain image, and that we assume that these fixation locations were generated by an IPP with log-intensity function $\eta(x, y)$.

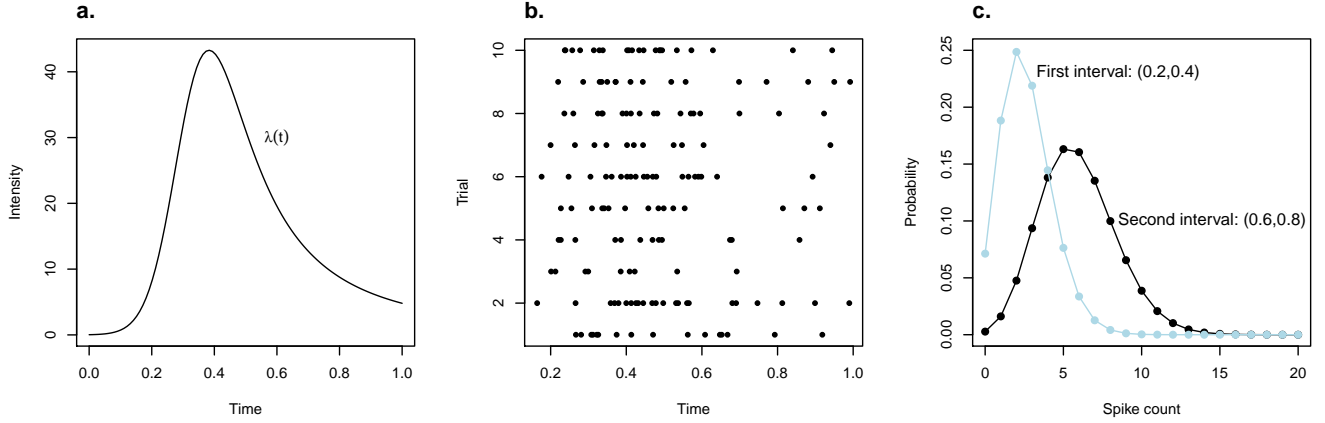


Figure 1: A first example of a point process: the Inhomogeneous Poisson Process (IPP) as a model for spike trains. **a.** The neuron is assumed to respond to stimulation at a varying rate over time. The latent rate is described by an intensity function, $\lambda(t)$ **b.** Spikes are stochastic: here we simulated spike trains from an IPP with intensity $\lambda(t)$. Different trials correspond to different independent realisations. Note that a given spike train can be seen simply as a set of points in $(0, 1)$. **c.** The defining property of the IPP is that spike counts in a given interval follow a Poisson distribution. Here we show the probability of observing a certain number of spikes in two different time intervals.

We might reasonably suppose that the value of $\eta(x, y)$ at location x, y has something to do with the local contrast $c(x, y)$ there. In other words, the image contrast function $c(x, y)$ will enter as a *covariate* in our model. The simplest way to do so is to posit that $\eta(x, y)$ is a linear function of $c(x, y)$, i.e.:

$$\eta(x, y) = \beta_c \times c(x, y) + \beta_0 \quad (2)$$

We have introduced two free parameters, β_c and β_0 , that will need to be estimated from the data. β_c is the more informative of the two: for example, a positive value indicates that high contrast is predictive of high intensity, and a nearly-null value would indicate that contrast is not related to intensity (or at least not monotonically). We will return to this idea below when we consider analysing the output of low-level saliency models.

Another example that will come up in our analysis is the well-documented issue of the “centrality bias”, whereby human observers in psychophysical experiments in front of a centrally placed computer screen tend to fixate central locations more often regardless of what they are shown (Tatler, 2007). Again this has an influence on the intensity function that needs to be accounted for. One could postulate another spatial (intrinsic) covariate, $d(x, y)$, representing the distance to the centre of the display: e.g., $d(x, y) = \sqrt{x^2 + y^2}$ assuming the centre is at $(0, 0)$. As in Equation (2), we could write

$$\eta(x, y) = \beta_d \times d(x, y) + \beta_0$$

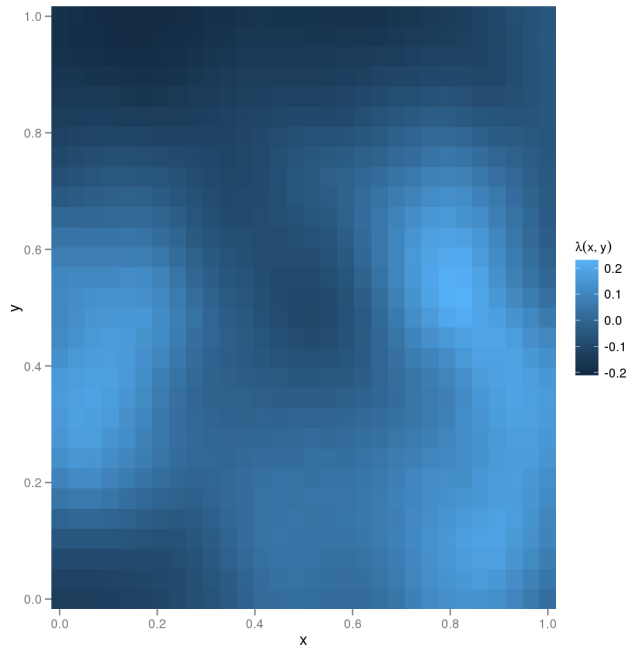
However, in a given image, both centrality bias and local contrast will play a role. It will then be sensible to take both into account, which leads naturally to:

$$\eta(x, y) = \beta_d \times d(x, y) + \beta_c \times c(x, y) + \beta_0 \quad (3)$$

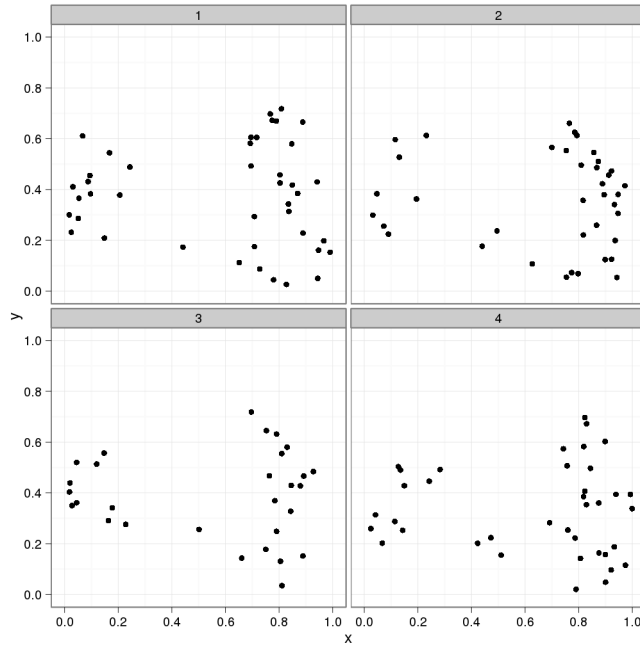
The relative contribution of each factor will be determined by the relative values of β_d and β_c . Such additive decompositions will be central to our analysis, and we will cover them in much more detail below.

2 Case study: Analysis of low-level saliency models

As already mentioned, a long-standing debate in the literature on attention has to do with the role of low-level image features in eye movement guidance. Very broadly speaking, the disagreement has to do with how *complex* and *flexible* eye-movement guidance really is: when moving our eyes, are we truly looking for objects, or rather relying on simple heuristics that often signal object boundaries (e.g, high local contrast, Nuthmann and Henderson, 2010) ? How far do we adjust our strategy according to task and context (Tatler et al., 2011)?



(a) The latent intensity function $\lambda(x, y)$ controls how many points will fall on average in a certain spatial area. Higher intensities are in lighter shades of blue.



(b) Four samples from an IPP with the intensity function shown in the left-hand panel.

Figure 2: The spatial IPP is a straightforward extension to the temporal IPP introduced in Figure 1. The main ingredient is a spatial intensity function $\lambda(x, y)$. The IPP produces random point sets, as in the right-hand panel. When analysing point data, the goal is usually to recover the latent intensity function from such samples.

We will focus on the question of low-level, local image cues. If eye movement guidance is a relatively inflexible system which uses local image cues as heuristics for finding interesting places in a stimulus, than low-level image cues should be predictive of where people look at when they have nothing particular to do. This has been investigated many times (see Schütz et al., 2011), and there are now many datasets available of “free-viewing” eye movements in natural images (Van Der Linde et al., 2009; Torralba et al., 2006). We use here the high-quality dataset of Kienzle et al. (2009), because the authors were particularly careful to eliminate a number of potential biases (photographer’s bias, among other things).

In Kienzle et al. (2009), subjects viewed photographs taken in a zoo in Southern Germany. Each image appeared for a short, randomly varying duration of around 2 sec¹. Subjects were instructed to “look around the scene”, with no particular goal given. The raw signal recorded from the eye-tracker was processed to yield a set of saccades and fixations, and here we focus only on the latter.

In such experiments, the question as usually formulated is to understand the structure of local image cues attracting fixations. Analysis often proceeds in a somewhat round-about way: one extracts image patches around fixated locations, and compares them to image patches around non-fixated locations to look for differences (Reinagel and Zador, 1999). An immediate benefit of our framework is that we can be a lot more direct and attempt to predict *where people will fixate in an image*. We will also choose to focus on the empty half of the glass and ask what it is that low level features *fail to predict*.

Why would that be interesting? In the context of the debate alluded to above, it has been claimed that low-level features are predictive of fixated locations only because the more sophisticated strategy at work happens to prefer locations that tend to *correlate* with certain local cues. This means in turn that there also should be important locations that low-level cues entirely fail to predict (as well as occasional “false alarms”), otherwise the two strategies would be exactly equivalent and one does not see why the brain should prefer the more complicated one.

2.1 Understanding the role of covariates in determining fixated locations

To be able to move beyond the basic statement that local image cues somehow *correlate* with fixation locations, it is important that we clarify how covariates could enter into the latent intensity function. There are many different ways in which this could happen, with important consequences for the modelling.

To begin with we imagine that local contrast is the only cue that matters. A very unrealistic but drastically simple model assumes that the more contrast there is in a region, the more subjects’ attention will be attracted to it. In our framework we could specify this model as:

$$\eta(x, y) = \beta_0 + \beta_1 c(x, y)$$

However, surely other things besides contrast matters - what about average luminance, for example? Couldn’t brighter regions attract gaze?

This would lead us to expand our model to include luminance as another spatial covariate, so that the log-intensity function would become:

$$\eta(x, y) = \beta_0 + \beta_1 c(x, y) + \beta_2 l(x, y)$$

in which $l(x, y)$ stands for local luminance. But perhaps edges matter, so why not include another covariate corresponding to the output of a local edge detector? We would have:

$$\eta(x, y) = \beta_0 + \beta_1 c(x, y) + \beta_2 l(x, y) + \beta_3 e(x, y)$$

It is of course possible to go further down this path, and add as many covariates as one sees fit (although with too many covariates, problems of variable selection do arise, see Hastie et al., 2003), but to make our lives simpler we can also rely on some prior work in the area and use pre-existing, off-the-shelf *image-based saliency models* (Fecteau and Munoz, 2006). Such models combine many local cues into one interest map, which saves us from having to choose a set of covariates and then estimating their relative weight. Here we focus on the perhaps most well-known among these models, described in Itti and Koch (2001) and Walther and Koch (2006), although many other interesting options are available (e.g., Bruce and Tsotsos, 2009, Zhao and Koch, 2011, or Kienzle et al., 2009).

The model computes several feature maps (orientation, contrast, etc.) according to physiologically plausible mechanisms, and combines them into one master map which aims to predict what the interesting features in image i are. For a given image i we can obtain the interest map $m_i(x, y)$ and use that as the unique covariate in a point process:

¹The actual duration was sampled from a Gaussian distribution $\mathcal{N}(2, 0.5^2)$ truncated at 1 sec.

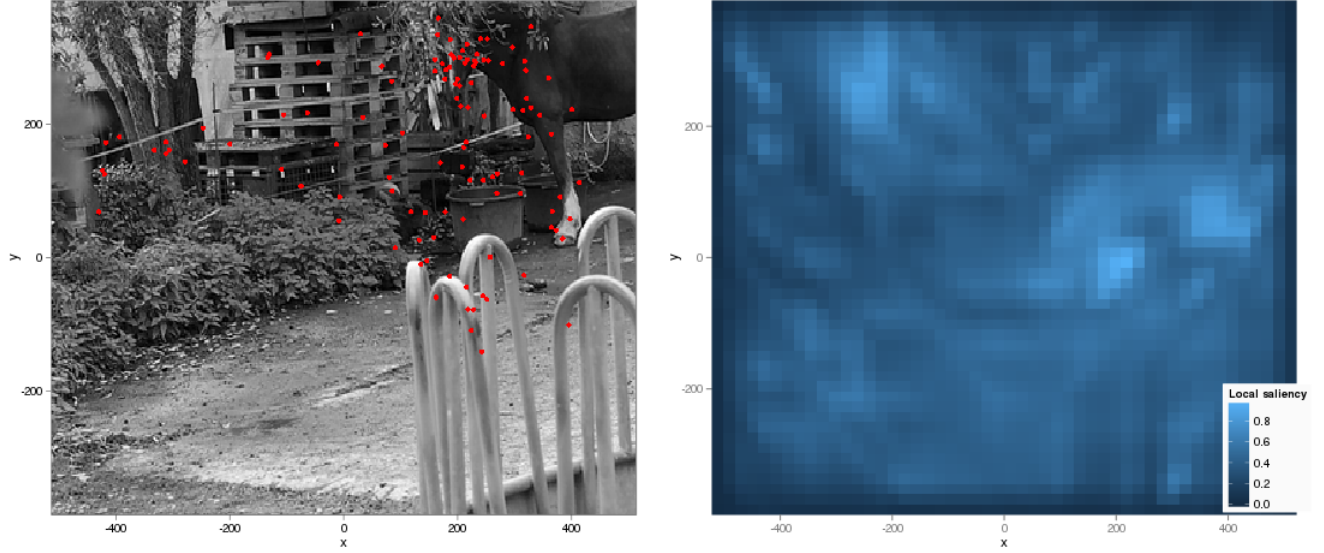


Figure 3: An image from the dataset of Kienzle et al. (2009), along with an “interest map” - local saliency computed according to the Itti-Koch model (Itti and Koch, 2001; Walther and Koch, 2006). Fixations made by the subjects are overlaid in red. How well does the interest map characterise this fixation pattern? This question is not easily answered by eye, but may be given a more precise meaning in the context of spatial processes.

$$\eta_i(x, y) = \alpha_i + \beta_i m_i(x, y) \quad (4)$$

This last equation will be the starting point of our modelling. We have changed the notation somewhat to reflect some of the adjustments we need to make in order to learn anything from applying model to data. To summarise:

- $\eta_i(x, y)$ denotes the log-intensity function for image i , which depends on the spatial covariate $m_i(x, y)$ that corresponds to the interest map given by the low-level saliency of Itti and Koch (2001).
- β_i is an image-specific coefficient that measures to what extent spatial intensity can be predicted from the interest map. $\beta_i = 0$ means no relation, $\beta_i > 0$ means that higher low-level saliency is associated on average with more fixations, $\beta_i < 0$ indicates the opposite - people looked more often at low points of the interest map. We make β_i image-dependent because we anticipate that how well the interest map predicts fixations depends on the image, an assumption that is borne out as we will see.
- α_i is an image specific intercept. It is required for technical reasons but plays otherwise no important role in our analysis.

We fitted the model given by Equation (4) to a dataset consisting of the fixations recorded in the first 100 images of the dataset of Kienzle et al. (2009). Computational methods are described in the appendix. We obtained a set of posterior estimates for the β_i 's, of which a summary is given in Figure 4.

To make the coefficients shown on Figure 4 more readily interpretable, we have scaled $m_i(x, y)$ so that in each image the most interesting points (according to the Itti-Koch model) have value 1 and the least interesting 0. In terms of the estimated coefficients β_i , this implies that the intensity ratio between a maximally interesting region and a minimally interesting region is equal to e^{β_i} : for example, a value of β_i of 1 indicates that in image i on average a region with high “interestingness” receives roughly 2.5 more fixations than a region with very low “interestingness”. At the opposite end of the spectrum, in images in which the IK model performs very well, we have values of $\beta_i \approx 6$, which implies a ratio of 150 to 1 for the most interesting regions compared to the least interesting.

It is instructive to compare the images in which the model does well², to those in which it does poorly. On Figure 5 we show the 8 images with highest β_i value, and on Figure 6 the 8 images with lowest β_i , along with the corresponding Itti-Koch interest maps. It is evident that, while on certain images the model does extremely well, for example when it

² β_i should not be interpreted as anything more than a rough measure of performance. It has a relatively subtle potential flaw: if the Itti-Koch map for an image happens by chance to match the typical spatial bias, then β_i will likely be estimated to be above 0. This flaw is corrected when a spatial bias term is introduced, see Section 2.4.

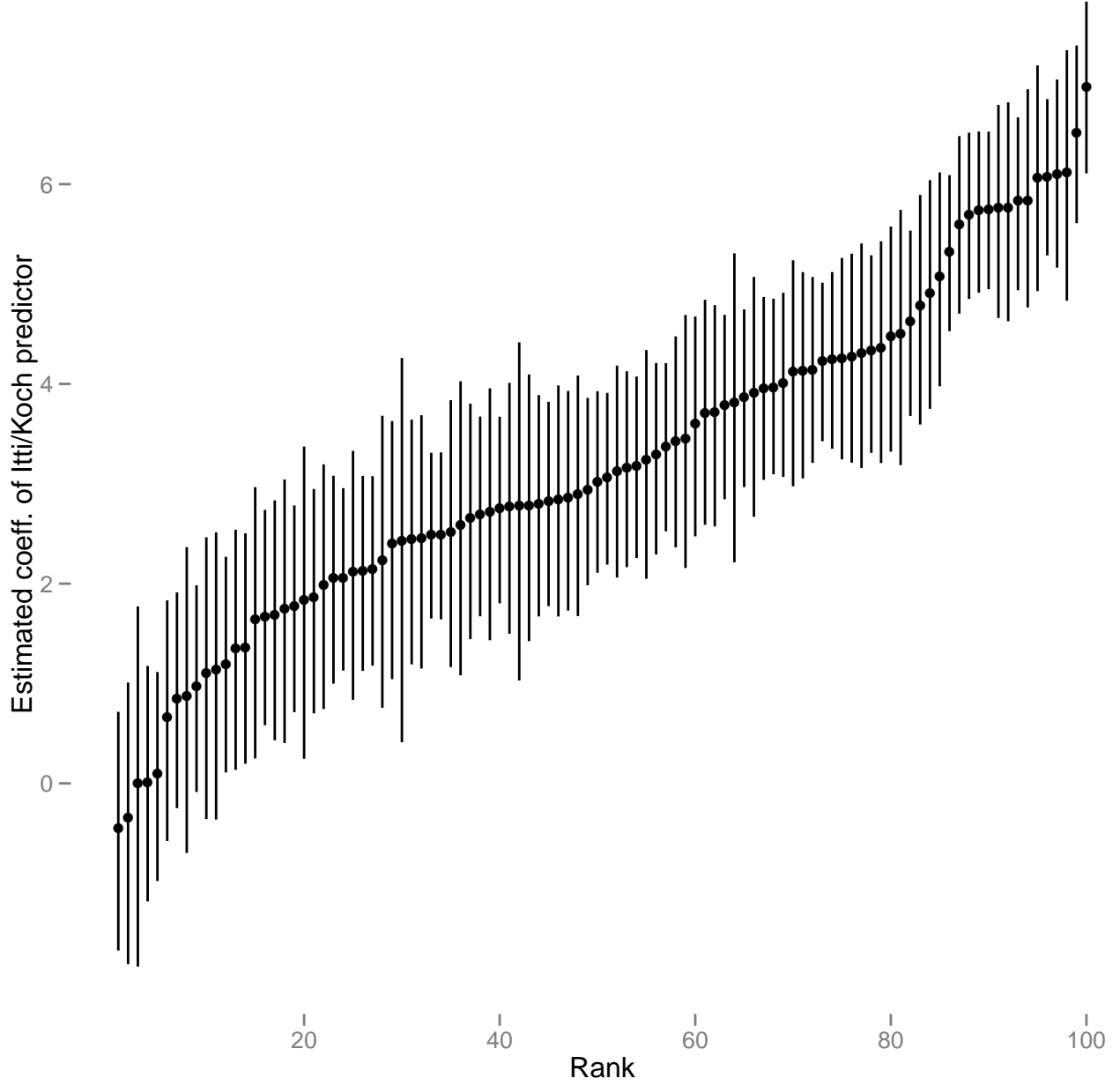


Figure 4: Variability in the predictivity of the Itti&Koch model across images. We estimate β_i in Equation 4 for 100 different images from the dataset of Kienzle et al. (2009). We plot the sorted mean-a-posteriori estimates along with a 95% Bayesian credible interval. The results show clearly that the “interestingness” given by low-level saliency is of variable value when predicting fixations: for some images β_i is very close to 0, which indicates that there is no discernible association between low-level saliency and fixation intensity in these images. In other images the association is much stronger.

manages to pick up the animal in the picture (see the lion in images 52 and 53), in others it gets fooled by high-contrast edges that subjects find highly uninteresting. Foliage and rock seem to be particularly difficult, at least from the limited evidence available here³.

Given a larger annotated dataset, it would be possible to confirm whether the model performs better for certain categories of images than others. Although this is outside the scope of the current paper, we would like to point out that the model in equation 4 can be easily extended for that purpose: If we assume that images are encoded as being either “foliage” or “not foliage”, we may then define a variable ϕ_i that is equal to 1 if image i is foliage and 0 if not. We may re-express the latent log-intensity as:

$$\eta_i(x, y) = \alpha_i + (\phi_i\gamma + \delta_i)m_i(x, y)$$

which decomposes β_i as the sum of an image-specific effect (δ_i) and an effect of belonging to the foliage category (γ). Having $\gamma < 0$ would indicate that pictures of foliage are indeed more difficult on average⁴.

A related suggestion (Torralla et al., 2006) is to augment low-level saliency models with some higher-level concepts, adding face detectors, text detectors, or horizon detectors. Within the limits of our framework, a much easier way to improve predictions is to take into account the *centrality bias* (Tatler and Vincent, 2009), i.e. the tendency for observers to fixate more often at the centre of the image than around the periphery. One explanation for the centrality bias is that it is essentially a side-effect of photographer’s bias: people are interested in the centre because the centre is where photographers usually put the interesting things, unless they are particularly incompetent. In Kienzle et al. (2009) photographic incompetence was simulated by randomly clipping actual photographs so that central locations were not more likely to be interesting than peripheral ones. The centrality bias persists (see Figure 7), which shows that central locations are preferred regardless of image content (a point already made in Tatler, 2007). We can use this fact to make better predictions by making the required modifications to the intensity function.

Before we can explain how to do that, we need to introduce a number of additional concepts. A central theme in the proposed spatial point process framework is to develop tools that help us to understand in detail the performance of our models. In the next section we introduce some relatively user-friendly graphical tools for assessing fit. We will also show how one can estimate an intensity function in a non-parametric way, that is, without assuming that the intensity function has a specific form. Nonparametric estimates are important in their own right for visualisation (see for example the right-hand-side of Figure 7), but also as a central element in more sophisticated analyses.

2.2 Graphical model diagnostics

One of the first things to do once one has fitted a statistical model to data is to make sure the fitted model is at least in rough agreement with the data. A good fit is naturally not the only thing we require of a model, because fits can in some cases be arbitrarily good if enough free parameters are introduced (see e.g., Bishop, 2007, ch. 3). Assessing fit is an important step in model criticism (Gelman and Hill, 2006), which will let us diagnose model failures, and in many cases will enable us to obtain a better understanding of the data itself. In this section we will focus on informal, graphical diagnostics. More advanced tools are described in Baddeley et al. (2005).

Since a statistical model is in essence a recipe for how the data are generated, the most obvious thing to do is to compare data simulated from the model to the actual data we measured. In the analysis presented above, the assumption is that the data come from a Poisson process whose log-intensity is a linear function of Itti-Koch interestingness:

$$\eta_i(x, y) = \alpha_i + \beta_i m_i(x, y) \tag{5}$$

For a given image, we have estimated values $\hat{\alpha}_i, \hat{\beta}_i$ (mean a posteriori estimate). A natural thing to do is to ask what data simulated from a model with those parameters look like⁵. In Figure 8, we take the image with the maximum estimated value for β_i and compare the actual recorded fixation locations to four different simulations from an IPP with the fitted intensity function.

What is immediately visible from the simulations is that, while the real data present one strong cluster that also appears in the simulations, the simulations have a higher proportion of points outside of the cluster, in areas far from any actual fixated locations. Despite these problems, the fit seems to be quite good compared to other examples from the dataset: figure 9 shows two other examples, image 45, which has a median β value of about 4, and image 32, which

³The full sorted list of images is available as supplementary material

⁴This may not necessarily be an intrinsic flaw of the model: it might well be that in certain “boring” pictures, or pictures with very many high-contrast edges, people will fixate just about anywhere, so that even a perfect model—the “true” causal model in the head of the observers—would perform relatively badly.

⁵Simulation from an IPP can be done using the “thinning” algorithm of Lewis and Shedler (1979), which is a form of rejection sampling.

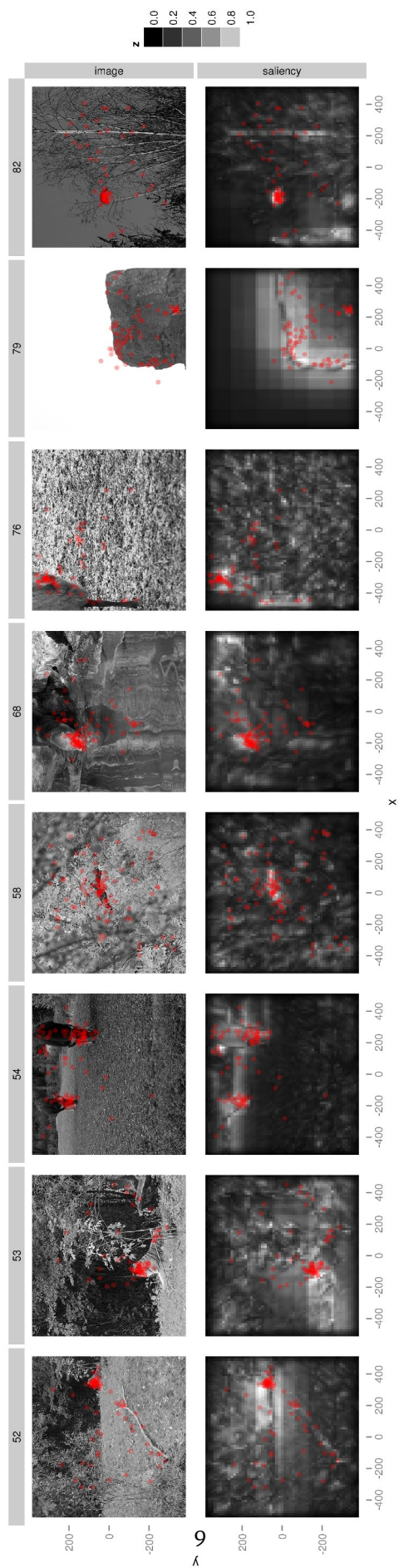


Figure 5: Out of first 100 images in the Kin8 dataset, we selected the 20 images which Ivis2V2 detected as

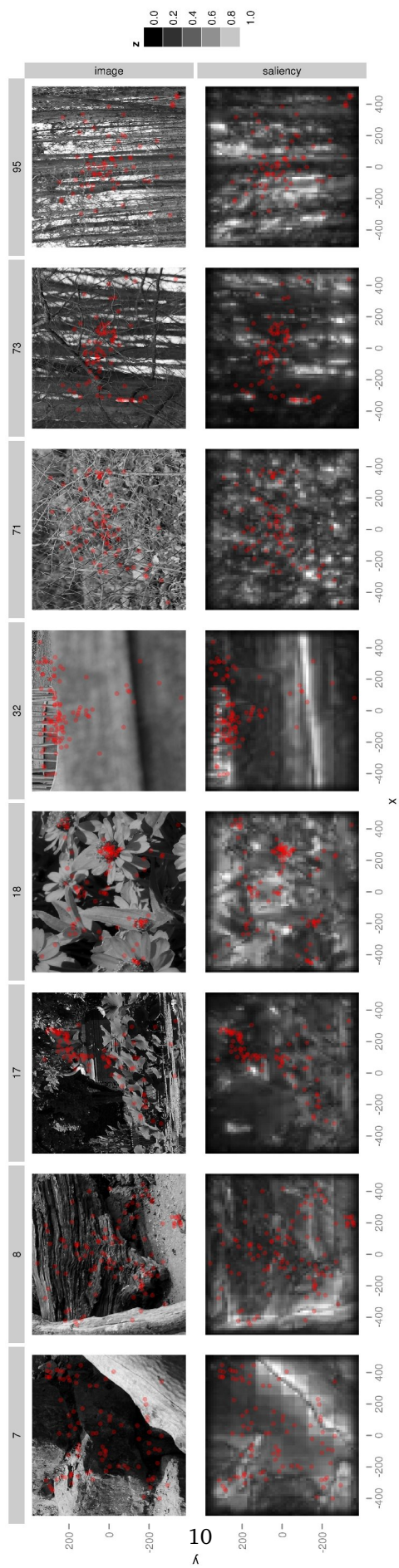


Figure 6.2: Figure 5b repeated with the 2 images with the highest saliency for $\lambda = 2$.

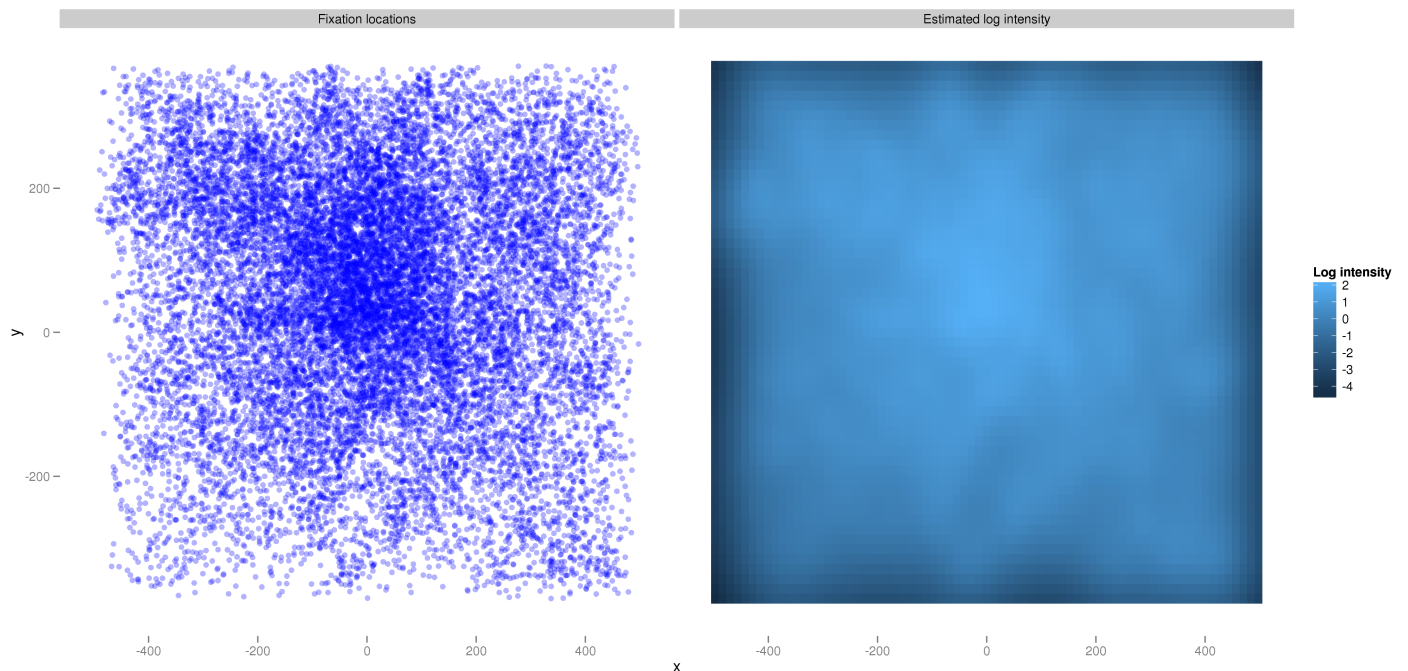


Figure 7: The centrality bias. On the left panel, we plot every fixation recorded in Kienzle et al. (2009). On the right, a non-parametric Bayesian estimate of the intensity function. Central locations are much more likely to be fixated than peripheral ones.

had a β value of about 0. In the case of image 32, since there is essentially no relationship between the interestingness values and fixation locations, the best possible intensity function of the form given by Equation (4) is one with $\beta = 0$, a uniform intensity function.

It is also quite useful to inspect some of the *marginal* distributions. By marginal distributions we mean point distributions that we obtain by merging data from different conditions. In figure 10, we plot the fixation locations across all images in the dataset. In the lower panel we compare it to simulations from the fitted model, in which we generated fixation locations from the fitted model for each image so as to simulate an entire dataset. This brings to light a failure of the model that would not be obvious from looking at individual images: based on Itti-Koch interestingness alone we would predict a distribution of fixation locations that is almost uniform, whereas the actual distribution exhibits a central bias, as well as a bias for the upper part of the screen.

Overall, the model derived from fitting Equation (4) seems rather inadequate, and we need to account at least for what seems to be some prior bias favouring certain locations. Explaining how to do so requires a short detour through the topic of non-parametric inference, to which we turn next.

2.3 Inferring the intensity function non-parametrically

Consider the data in Figure 7: to get a sense of how much observers prefer central locations relative to peripheral ones, we could define a central region \mathcal{A} , count how many fixations fall in it, compared to how many fixations fall outside. From the theoretical point of view, what we are doing is directly related to estimating the intensity function: the expected number of fixations in \mathcal{A} is after all $\int_{\mathcal{A}} \lambda(x, y) dx dy$, the integral of the intensity function over \mathcal{A} . Seen the other way, counting how many sample points are in \mathcal{A} is a way of estimating the integral of the intensity over \mathcal{A} .

Modern statistical modelling emphasizes non-parametric estimation. If one is trying to infer the form of an unknown function $f(x)$, one should not assume that $f(x)$ has a certain parametric form unless there is very good reason for this choice (interpretability, actual prior knowledge or computational feasibility). Assuming a parametric form means assuming for example that $f(x)$ is linear, or quadratic: in general it means assuming that $f(x)$ can be written as $f(x) = \phi(x; \beta)$, where β is a finite set of unknown parameters, and $\phi(x; \beta)$ is a family of functions over x parameterised

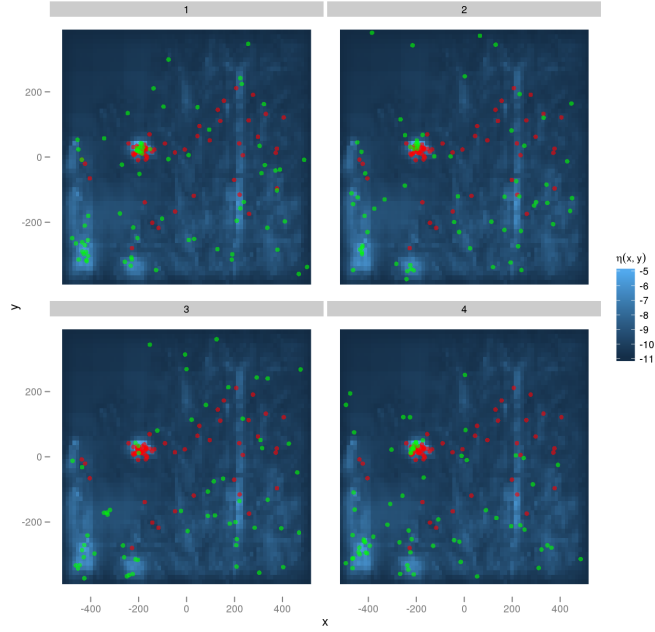


Figure 8: Simulating from a fitted point process model. The fixations on image 82 (rightmost in Figure 5) were fitted with the model given by Equation (4), resulting in an estimated log-intensity $\eta(x, y)$ which is plotted as a heatmap in the background of each panel. In red we plot the actual fixation locations, and in green simulations from the fitted model. We show four independent simulations (only the green points vary).

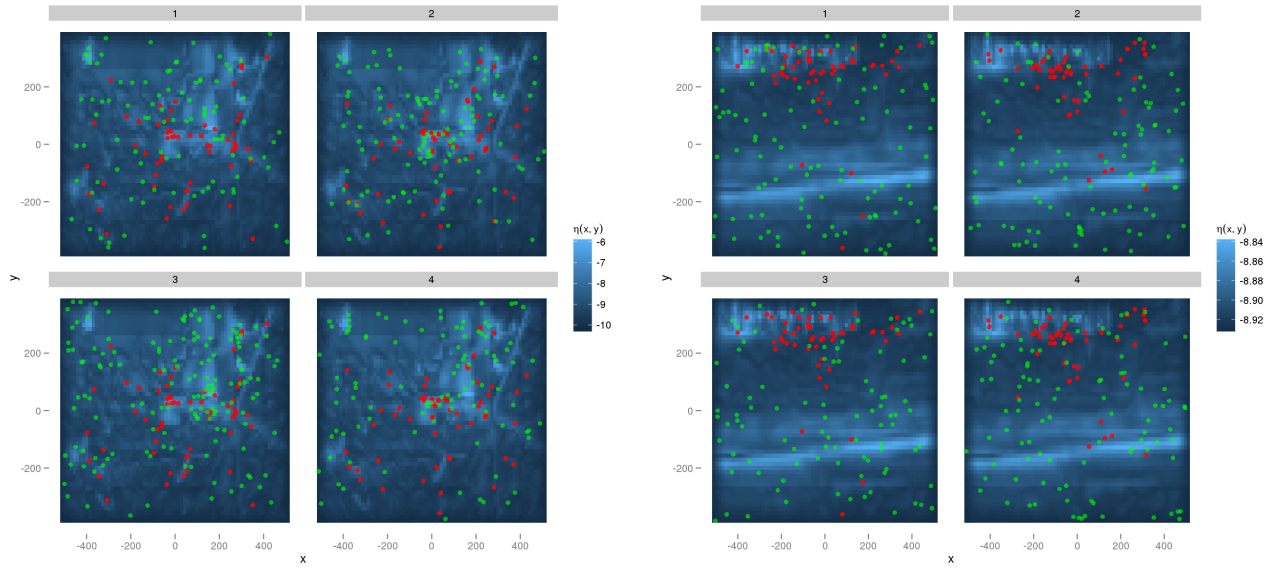


Figure 9: Same as in figure 8, with the fixations measured on image 45 (left) and 32 (right) of the dataset. The agreement between data and simulations is of distinctly poorer quality than in image 82.

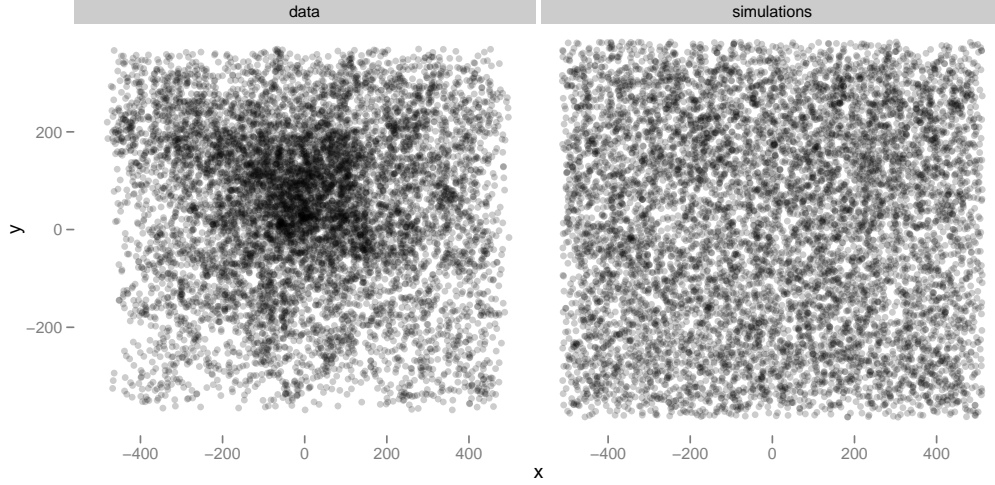


Figure 10: Comparing marginal fixation locations. On the left panel, we plot each fixation location in images 1 to 100 (each dot corresponds to one fixation). On the right panel, we plot simulated fixation locations from the fitted model corresponding to Equation (4). A strong spatial bias is visible in the data, not captured at all by the model.

by β . Nonparametric methods make much weaker assumptions, usually assuming only that f is smooth at some spatial scale.

We noted above that estimating the integral of the intensity function over a spatial region could be done by counting the number of points the region contains. Assume we want to estimate the intensity $\lambda(x, y)$ at a certain point x_0, y_0 . We have a realisation S of the point process (for example a set of fixation locations). If we assume that $\lambda(x, y)$ is spatially smooth, it implies that $\lambda(x, y)$ varies slowly around x_0, y_0 , so that we may consider it roughly constant in a small region around x_0, y_0 , for instance in a circle of radius r around (x_0, y_0) . Call this region C_r - the integral of the intensity function over C_r is related to the intensity at (x_0, y_0) in the following way:

$$\int_{C_r} \lambda(x, y) dx dy \approx \int_{C_r} \lambda(x_0, y_0) dx dy = \lambda(x_0, y_0) \times \int_{C_r} dx dy$$

$\int_{C_r} dx dy$ is just the area of circle C_r , equal to πr . Since we can estimate $\int_{C_r} \lambda(x, y) dx dy$ via the number of points in C_r , it follows that we can estimate $\lambda(x_0, y_0)$ via:

$$\hat{\lambda}(x_0, y_0) = \frac{|S \cap C_r|}{\pi r}$$

$|S \cap C_r|$ is the cardinal of the intersection of the point set S and the circle C_r (note that they are both sets), shorthand for “number of points in S that are also in C_r ”.

What we did for (x_0, y_0) remains true for all other points, so that a valid strategy for estimating $\lambda(x, y)$ at any point is to count how many points in S are in the circle of radius r around the location. The main underlying assumption is that $\lambda(x, y)$ is roughly constant over a region of radius r . This method will be familiar to some readers in the context of non-parametric density estimation, and indeed it is almost identical⁶. It is a perfectly valid strategy, detailed in Diggle (2002), and its only major shortcoming is that the amount of smoothness (represented by r) one uses changes the results quite dramatically (see Figure 11). Although it is possible to also estimate r from the data, in practice this may be difficult (see Illian et al., 2008, section 3.3).

There is a Bayesian alternative: put a prior distribution on the intensity λ and base the inference on the posterior distribution of $\lambda(x, y)$ given the data, with

$$p(\lambda|S) \propto p(S|\lambda)p(\lambda)$$

as usual. We can use the posterior expectation of $\lambda(x, y)$ as an estimator (the posterior expectation is the mean value of $\lambda(x, y)$ given the data), and the posterior quantiles give confidence intervals⁷. The general principles of Bayesian

⁶Most often, instead of using a circular window, a Gaussian kernel will be used.

⁷For technical reasons Bayesian inference is easier when done on the log-intensity function $\eta(x, y)$, rather than on the intensity function, so we actually use the posterior mean and quantiles of $\eta(x, y)$ rather than that of $\lambda(x, y)$.

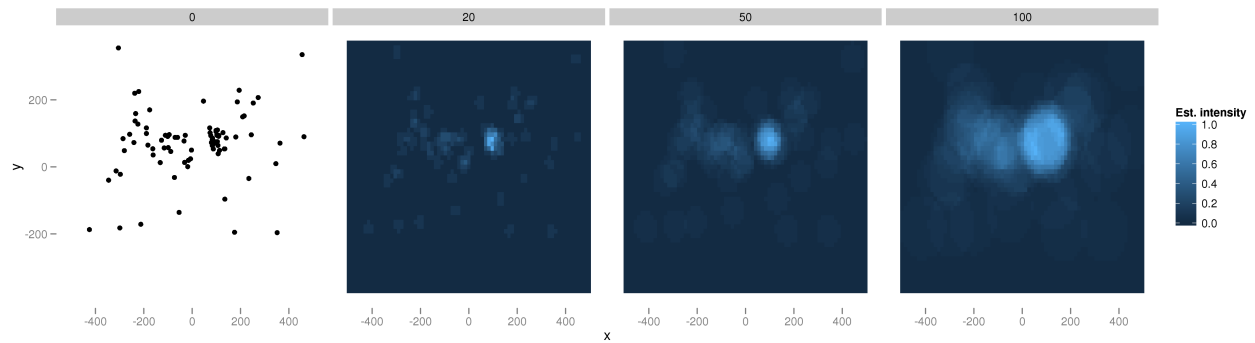


Figure 11: Nonparametric estimation of the intensity function using a moving window. The data are shown on the leftmost panel. The intensity function at (x, y) is estimated by counting how many points are within a radius r of (x, y) . We show the results for $r = 20, 50, 100$. Note that with $r \rightarrow 0$ we get back the raw data. For easier visualization, we have scaled the intensity values such that the maximum intensity is 1 in each panel.

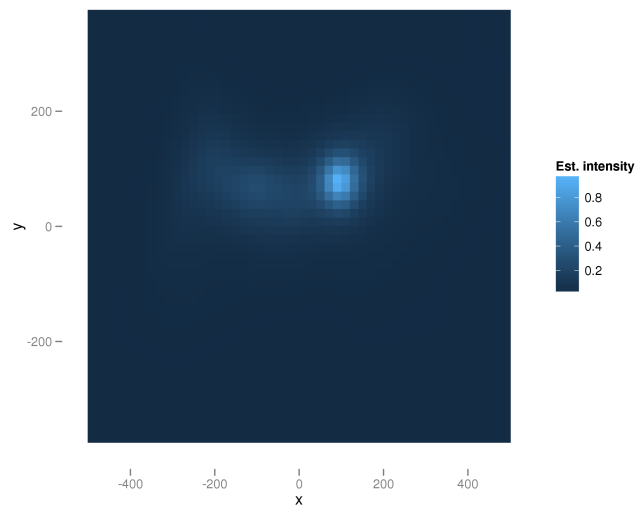


Figure 12: Nonparametric Bayesian estimation of the intensity function. We use the same data as in Figure 11. Inference is done by placing a Gaussian process prior on the log-intensity function, which enforces smoothness. Hyperparameters are integrated over. See text and appendix 4.2 for details.

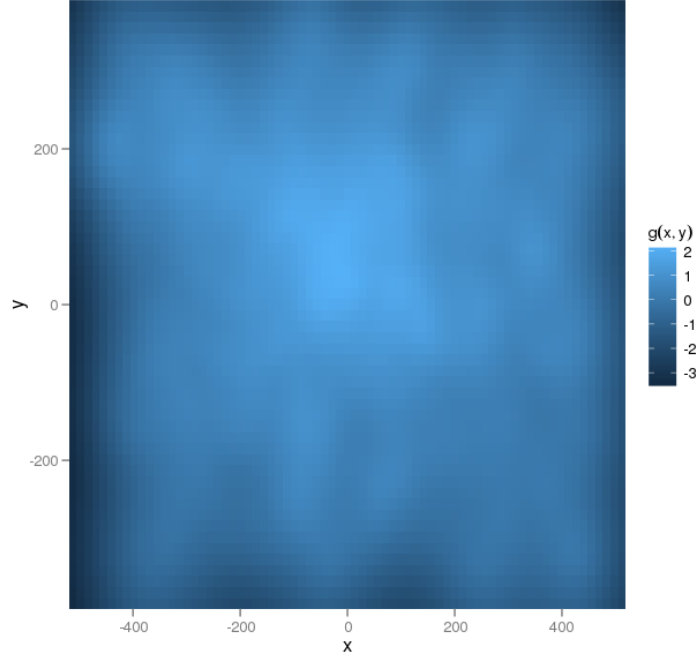


Figure 13: Estimated spatial bias $g(x, y)$ (Equation 6).

statistics will not be explained here, the reader may refer to Kruschke (2010) for an introduction.

To be more precise, the method proceeds by writing down the very generic model:

$$\log \lambda(x, y) = f(x, y) + \beta_0$$

and effectively forces $f(x, y)$ to be a relatively smooth function, using a Gaussian Process prior. Exactly how this is achieved is explained in Appendix 4.2, but roughly, Gaussian Processes let one define a probability distribution over functions such that smooth functions are much more likely than non-smooth functions. The exact spatial scale over which the function is smooth is unknown but can be averaged over.

To estimate the intensity function of one individual point process, there is little cause to prefer the Bayesian estimate over the classical non-parametric estimate we described earlier. As we will see however, using a prior that favours smooth functions becomes invaluable when one considers *multiple point processes* with shared elements.

2.4 Including a spatial bias, and looking at predictions for new images

We have established that models built from interest maps do not fit the data very well, and we have hypothesized that one possible cause might be the presence of a spatial bias. Certain locations might be fixated despite having relatively uninteresting contents. A small modification to our model offers a solution: we can hypothesize that all latent intensity functions share a common component. In equation form:

$$\eta_i(x, y) = \alpha_i + \beta_i m_i(x, y) + g(x, y) \quad (6)$$

As in the previous section, we do not force $g(x, y)$ to take a specific form, but only assume smoothness. Again, we use the first 100 images of the dataset to estimate the parameters. The estimated spatial bias is shown on fig 13. It features the centrality bias and the preference for locations above the midline that were already visible in the diagnostics plot of Section 2.2 (Figure 10).

From visual inspection alone it appears clear that including a spatial bias is necessary, and that the model with spatial bias offers a significant improvement over the one that does not. However, things are not always as clear-cut, and one cannot necessarily argue from a better fit that one has a better model. There are many techniques for statistical model comparison, but given sufficient data the best is arguably to compare the predictive performance of the different models in the set. In our case we could imagine two distinct prediction scenarios:

1. For each image, one is given, say, 80% of the recorded fixations, and must predict the remaining 20%.

2. One is given all fixation locations in the first n images, and must predict fixations locations in the next k .

To use low-level saliency maps in engineering applications (Itti, 2004), what is needed is a model that predicts fixation locations on arbitrary images—i.e. the model needs to be good at the second prediction scenario outlined above. The model is tuned on recorded fixations on a set of training images, but to be useful it must make sensible predictions for images outside the original training set.

From the statistical point of view, there is a crucial difference between prediction problems (1) and (2) above. Problem (1) is easy: to make predictions for the remaining fixations on image i , estimate β_i and α_i from the available data, and predict based on the estimated values (or using the posterior predictive distribution). Problem (2) is much more difficult: for a new image j we have no information about the values of β_j or α_j . In other words, in a new image the interest map could be very good or worthless, and we have no way of knowing that in advance.

We do however have information about what values β_j and α_j are likely to take from the estimated values for the images in the training set. If in the training set nearly all values $\beta_1, \beta_2, \dots, \beta_n$ were above 0, it is unlikely that β_j will be negative. We can represent our uncertainty about β_j with a probability distribution, and this probability distribution may be estimated from the estimated values for $\beta_1, \beta_2, \dots, \beta_n$. We could, for example, compute their mean and standard deviation, and assume that β_j is Gaussian distributed with this particular mean and standard deviation⁸. Another way, which we adopt here, is to use a kernel density estimator so as not to impose a Gaussian shape on the distribution.

As a technical aside: for the purpose of prediction the intercept α_j can be ignored, as its role is to modulate the intensity function globally, and it has no effect on where fixations happen, simply on *how many* fixations are predicted. Essentially, since we are interested in fixation locations, and not in how many fixations we get for a given image, we can safely ignore α_i . A more mathematical argument is given in Appendix 4.3.2.

Thus how to predict? We know how to predict fixation locations *given* a certain value of β_j , as we saw earlier in Section 2.2. Since β_j is unknown we need to average over our uncertainty. A recipe for generating predictions is to sample a value for β_j from $p(\beta_j)$, and conditional on that value, sample fixation locations. Please refer to Figure 14 for an illustration.

In Figure 15 we compare predictions for marginal fixation locations (over all images), with and without a spatial bias term. We simulated fixations from the predictive distribution for images 101 to 200. We plot only one simulation, since all simulations yield for all instance and purposes the same result: without a spatial bias term, we replicate the problem seen in Figure 10. We predict fixations distributed more or less uniformly over the monitor. Including a spatial bias term solves the problem.

What about predictions for individual images? Typically in vision science we are attempting to predict a one-dimensional quantity: for example, we might have a probability distribution for somebody's contrast threshold. If this probability distribution has high variance, our predictions for any *individual* trial or the average of a number of trials are by necessity imprecise. In the one-dimensional case it is easy to visualise the degree of certainty by plotting the distribution function, or providing a confidence interval. In a point process context, we do not deal with one-dimensional quantities anymore: if the goal is to predict where 100 fixations on image j might fall, we are dealing with a 200 dimensional space—100 points times 2 spatial dimensions. A maximally confident prediction would be represented by a probability distribution that says that all points will be at a single location, e.g. (55, 235). A minimally confident prediction would be represented by the uniform distribution over the space of possible fixations, saying that all possible configurations are equally likely. Thus the question that needs to be addressed is, where do the predictions we can make from the Itti-Koch model fall along this axis?

It is impossible to provide a probability distribution, or to report confidence intervals. A way to visualise the amount of uncertainty we have is by drawing samples from the predictive probability distribution, to see if the samples vary a lot. Each sample is a set of a 100 points: if we notice for example that over 10 samples all the points in each sample systematically cluster at a certain location, it indicates that our predictive distribution is rather specific. If we see a lot of variability across samples, it is not. This mode of visualisation is better adapted to a computer screen than to be printed on paper, but for a few examples we attempt it anyway as shown in Figure 16.

To better understand the level of uncertainty involved, imagine that the objective is to perform (lossy) image compression. Image compression works by discarding information and hoping people will not notice. The promise of

⁸There is a cleaner way of doing that, using multilevel/random effects modelling (Gelman and Hill, 2006), but a discussion of these techniques would take us outside the scope of this work.

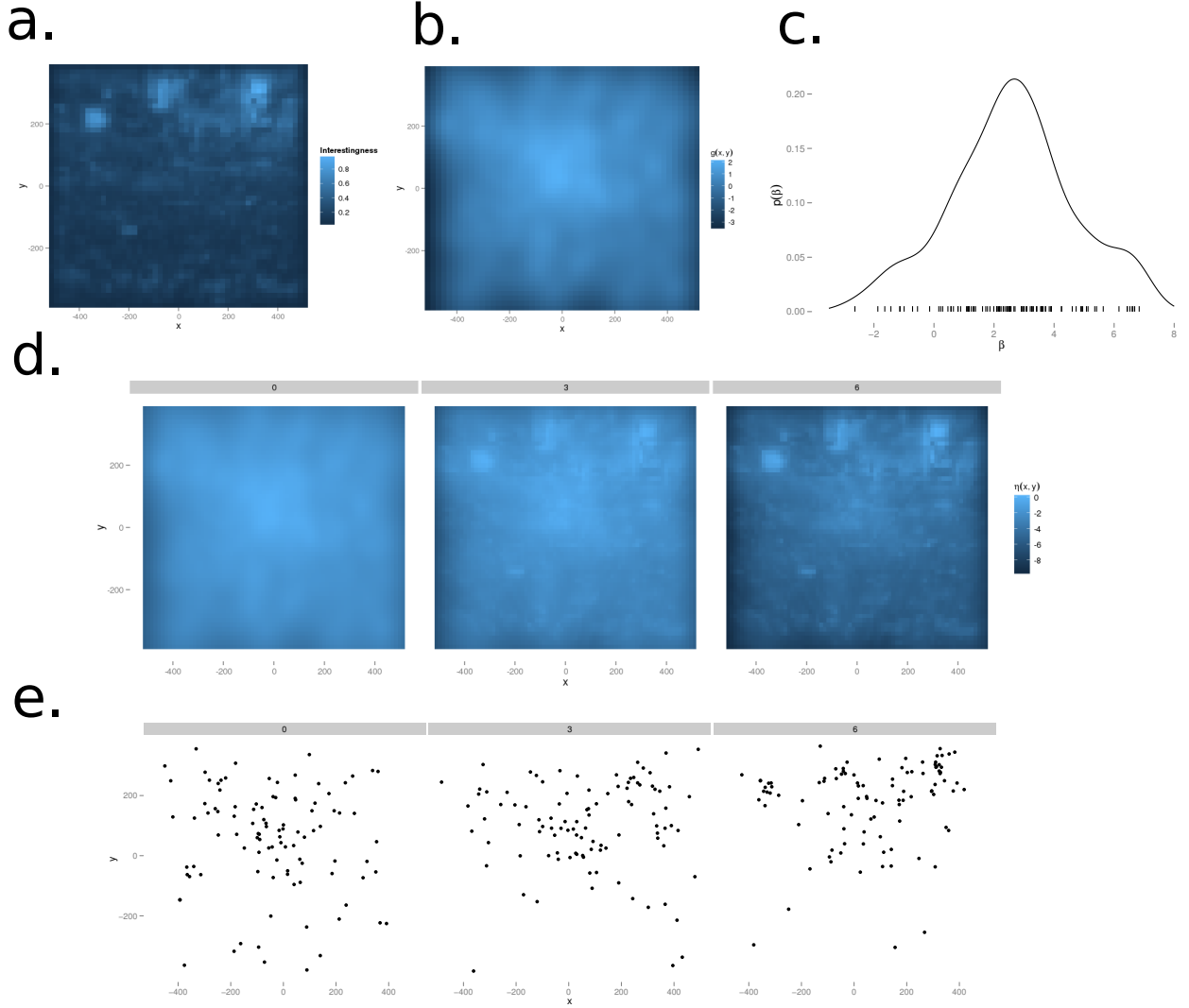


Figure 14: Predictions for a novel image. The latent intensity function in equation 6 has two important components: the interest map $m(x, y)$, shown here in panel (a) for image 103, and a general spatial component $g(x, y)$, shown in (b). Image 103 does not belong to the training set, and the value of β_{103} is therefore unknown: we do not know if $m(x, y)$ will be a strong predictor or not, and must therefore take this uncertainty into account. Uncertainty is represented by the distribution the β coefficient takes over images, and we can estimate this distribution from the estimated values from the training set. In (c) we show those values as dashes, along with a kernel density estimate. Conditional on a given value for β_{103} , our predictions come from a point process with log-intensity function given by $\beta_{103}m_{103}(x, y) + g(x, y)$: in (d), we show the intensity function for $\beta_{103} = 0, 3, 6$. In (e), we show simulations from the corresponding point processes (conditional on $n = 100$ fixations, see 4.3.2). In general the strategy for simulating from the predictive distribution will be to sample a value of β from $p(\beta)$, and sample from the corresponding point process as is done here.

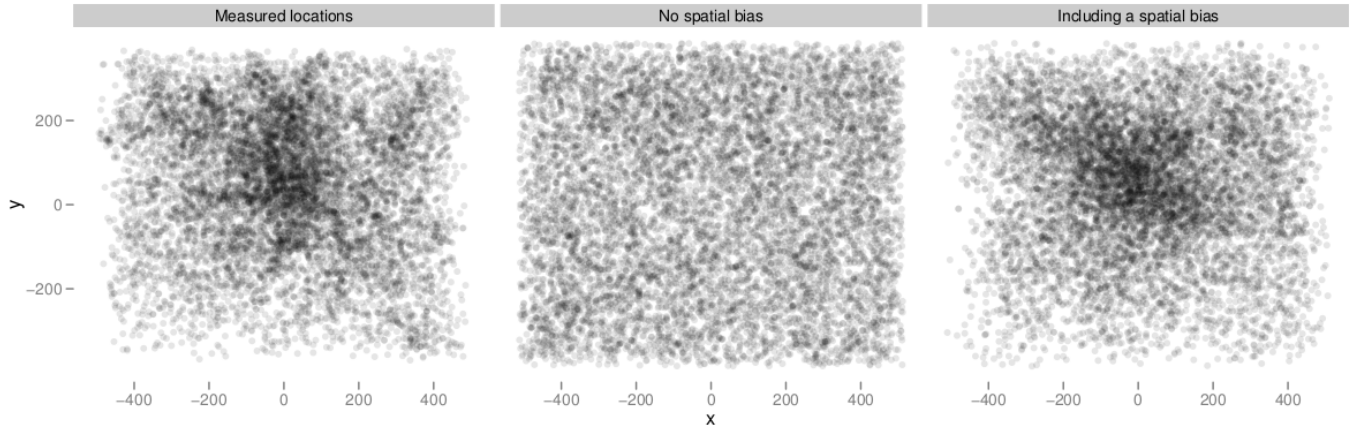


Figure 15: Predicting marginal fixation locations. In the first panel we plot the location of all fixations for images 101 to 200. In the second panel we plot simulated fixation locations for the same images from the naive model of Equation 4. In the second panel we plot simulated fixation locations for the same images from the model of Equation 13, which includes a spatial bias. Note that these are predicted fixation locations for entirely new images, and not a fit. Including a spatial bias improves predictions enormously.

image-based saliency models is that if we can predict what part of an image people find interesting, we can get away with discarding more information where people will not look. Let us simplify the problem and assume that either we compress an area or we do not. The goal is to find the largest possible section of the image we can compress, under the constraint that if a 100 fixations are made in the image, less than 5 fall in the compressed area (with high probability). If the predictive distribution is uniform, we can afford to compress less than 5% of the area of the image. A full formalisation of the problem for other distributions is rather complicated, and would carry us outside the scope of this introduction, but looking at the examples of Figure 16 it is not hard to see qualitatively that for most images, the best area we can find will be larger than 5% but still rather small: in the predictive distributions, points have a tendency of falling in most places except around the borders of the screen.

The reason we see such imprecision in the predictive distributions is essentially because we have to hedge our bets: since the value of β may vary a lot our predictions are vague by necessity. The conclusions of the previous section apply here again. We have essentially two ways forward: one is to enrich the model by increasing the number of spatial covariates used. The other is to see if the distribution of β changes across image categories, so that we may be able to make more precise predictions for certain image categories than others. We leave these two suggestions for future work.

3 Discussion

We hope to have convinced the reader that the point of view of spatial point processes is a fruitful one for researchers interested in eye movements. They bring clarity and precision to our thinking about what it means to predict eye movements, how models should be evaluated, how covariates can be used, etc. In the example we chose here, looking at a model by Itti and Koch (2001) of low-level saliency, we were able to show that although the model had predictive value on average, it had varying usefulness from one image to another. We believe that the consequences of this problem for prediction are under-appreciated: as we stated in the last section, when trying to predict fixations over an arbitrary image, this variability in quality of the predictor leads to predictions that are necessarily vague. Although insights like this one could be arrived at starting from other viewpoints, they arise very naturally from the framework presented here.

We need to stress that the kind of modelling we have done here does not address causality. The fact that fixation locations can be predicted from a certain spatial covariate does not imply that the spatial covariate causes the apparition of points. To take a concrete example, one can probably predict the world-wide concentration of polar bears from the quantities of ice-cream sold, but that does not imply that low demand for ice-cream causes polar bears to appear. The same caveat apply in spatial point process models as in regression modelling, see Gelman and Hill (2006).

Naturally, there is much we have left out, and we would like to raise some of the remaining issues in the following. First, we have left the temporal dimension completely out of the picture. Adding a temporal dimension in point process

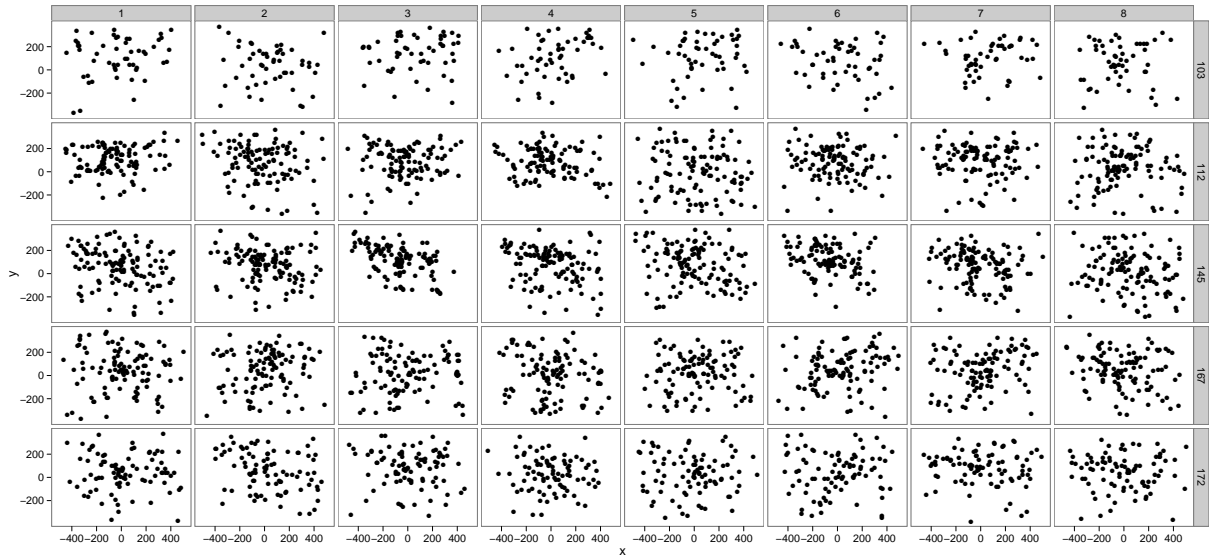


Figure 16: Samples from the predictive distributions for the model including spatial bias. We picked five images at random, and generated 8 samples from the predictive distribution from each, using the technique outlined in Figure 14. Each row corresponds to one image, with samples along the columns. This lets us visualise the uncertainty in the predictive distributions, see text.

models presents no conceptual difficulty; and we could extend the analyses presented here to see in detail whether, for example, low-level saliency predicts earlier fixations better than later ones. Unfortunately, although these extensions are conceptually simple, computational problems can be relatively severe for large datasets. One runs into memory limitations, enormous computing times, and often both. The silver lining is that there is active work in the area from computational statisticians, and we can expect swift progress.

Second, in this work we have considered that a fixation is nothing more than a dot: it has spatial coordinates and nothing more. Of course, this is not true: a fixation lasted a certain time, during which particular fixational eye movements occurred, etc. Studying fixation duration is an interesting topic in its own right, because how long one fixates might be tied to the cognitive processes at work in a task (Nuthmann et al., 2010). There are strong indications that when reading, gaze lingers longer on parts of text that are harder to process. Among other things, the less frequent a word is, the longer subjects tend to fixate it (Kliegl et al., 2006). Saliency is naturally not a direct analogue of word frequency, but one might nonetheless wonder whether interesting locations are also fixated longer. We could take our data to be fixations coupled with their duration, and we would have what is known in the spatial statistics literature as a *marked point process*. Marked point processes could be of extreme importance to the analysis of eye movements, and we refer the reader to Illian et al. (2009) for some ideas.

Third, another limitation we need to state is that the point process models we have described here do not deal very well with high measurement noise. We have assumed that what is measured is an actual fixation location, and not a noisy measurement of an actual measurement location. In experiments in which eye movements are measured with a high-quality tracker, and in which calibration is regularly tested, we believe that this assumption is not a problem. Issues do arise when the scale of measurement error is larger than the typical scale at which spatial covariates change. Although there are theoretical solutions to this problem (involving mixture models), they are rather cumbersome from a computational point of view. An less elegant work-around is to blur the covariates at the scale of measurement error.

Finally there is the practical question of the availability of user-friendly tools for point process analysis. The **spatstat** package in R (Baddeley and Turner, 2005) is very complete and well-documented, but its focus on single point processes makes it insufficient for the analyses shown here. We are in the process of writing a R package of our own, tentatively called **mpp**, for “multiple point processes”, to fill this gap. Some preliminary code will soon be made available on the first author’s webpage, which should be enough to reproduce most of the figures in this paper.

4 Appendices

4.1 The messy details of spatial statistics, and how to get around them

Spatial statistics is a treasure chest of useful concepts for the analysis of eye movements, but is sadly not the most accessible. There are two main hurdles. One is that the traditional application fields (ecology, forestry, epidemiology) have a rather separate set of problems from ours. Textbooks, such as the excellent one by Illian et al. (2008), focus on non-Poisson processes because applied problems often have to do with how far trees are from one another and whether bison are more likely to be in groups of three than all by themselves. Such questions have to do with the second-order properties of point processes, which express how much points are likely to attract or repel one another, and are not so central to the analysis of fixations (except, perhaps, to model the so-called inhibition-of-return). Unfortunately, formulation point process models with interesting second-order properties requires rather sophisticated mathematics, which makes the material less accessible than it could be.

The second issue is while the formal properties of point process models are well-known, practical use is hindered by computational difficulties. A very large part of the literature focuses on computational techniques (maximum likelihood or Bayesian) for fitting point process models. Much progress has been made recently (see, among others, Haran and Tierney, 2012, or Rue et al., 2009) but difficulties remain. We doubt that the nature of these difficulties would be of direct interest to most eye movement researchers, so we have tried to build a toolkit for the R environment that attempts to sweep most nasty details under the carpet. We build on one of the best techniques available (INLA) to provide a generic way to fit multiple point process models without worrying too much about the mechanics. The toolkit will soon be available for download on the first author's webpage, and a technical report describing its usage is being prepared.

4.2 Gaussian processes and Gauss-Markov processes

Gaussian Processes (GPs) and related methods are tremendously useful but not the easiest to explain. We will stay here at a conceptual level, computational details can be found in the monograph of Rasmussen and Williams (2005).

Rather than directly state how we use GPs in our models, we start with a detour on non-parametric regression (see Figure 18), which is where Gaussian processes are most natural. In non-parametric regression, given the (noisy) values of a function $f(x)$ measured at points x_1, \dots, x_n , we try to infer what the values of f are at other points. *Interpolation* and *extrapolation* can be seen as special, cases of non-parametric regression - ones where noise is negligible. The problem is non-parametric because we do not wish to assume that $f(x)$ has a known parametric form (for example, that f is linear).

For a statistical solution to the problem, we need a likelihood, and usually it is assumed that $y_i|x_i \sim \mathcal{N}(f(x_i), \sigma^2)$ which corresponds to observing the true value corrupted by Gaussian noise of variance σ^2 . This is not enough, since there are uncountably many functions f that have the same likelihood, namely all those that have the same value at the sampling points x_1, \dots, x_n (fig 18).

Thus, we need to introduce some constraints. Parametric methods constrain f to be in a certain class, and can be thought of as imposing “hard” constraints. Nonparametric methods such as GP regression impose *soft* constraints, by introducing an a priori probability on possible functions such that reasonable functions are favoured (figure 18 and 19). In a Bayesian framework, this works as follows. What we are interested in is the posterior probability of f given the data, which is as usual given by $p(f|\mathbf{y}) \propto p(\mathbf{y}|f)p(f)$. As we mentioned above $p(\mathbf{y}|f) = \prod_{i=1}^n \mathcal{N}(y_i|f(x_i), \sigma^2)$ is equal for all functions that have the same values at the sampled points x_1, \dots, x_n , so what distinguishes them in the posterior is how likely they are a priori—which is, of course, provided by the prior distribution $p(f)$.

How to formulate $p(f)$? We need a probability distribution that is defined over a space of functions. The idea of a process that generates random functions may not be as unfamiliar as it sounds: a Wiener process, for example, can be interpreted as generating random functions (Figure 17a). A Wiener process is a diffusion: it describes the random motion of a particle over time. To generate the output of a Wiener process, you start at time t_0 with a particle at position $z(t_0)$, and for each infinitesimal time increment you move the particle by a random offset, so that over time you generate a “sample path” $z(t)$.

This sample path might as well be seen as a function, just like the notation $z(t)$ indicates, so that each time one runs a Wiener process, one obtains a different function. This distribution will probably not have the required properties for most applications, since samples from a Wiener process are much too noisy - they generate functions that look very rough and jagged. The Wiener process is however a special case of a GP, and this more general family has some much more nicely-behaved members.

A useful viewpoint on the Wiener process is given by how successive values depend on each other. Suppose we simulate many sample paths of the WP, and each time measure the position at time t_a and t_b , so that we have a

collection of m samples $\{(z_1(t_a), z_1(t_b)), \dots, (z_m(t_a), z_m(t_b))\}$. It is clear that $z(t_a)$ and $z(t_b)$ are not independent: if t_a and t_b are close, then $z(t_a)$ and $z(t_b)$ will be close too. We can characterise this dependence using the covariance between these two values: the higher the covariance, the more likely $z(t_a)$ and $z(t_b)$ are to be close in value. Figure 17b illustrates this idea.

If we could somehow specify a process such that the correlation between two function values at different places does not decay too fast with the distance between these two places, then presumably the process would generate rather smooth functions. This is exactly what can be achieved in the GP framework. The most important element of a GP is the covariance function $k(x, x')$ ⁹, which describes how the covariance between two function values depend on where the function is sampled: $k(x, x') = \text{Cov}(f(x), f(x'))$.

We now have the necessary elements to define a GP formally. A GP with mean 0 and covariance function $k(x, x')$ is a distribution on the space of functions of some input space \mathcal{X} into \mathbb{R} , such that for every set of $\{x_1, \dots, x_n\}$, the sampled values $f(x_1), \dots, f(x_n)$ are such that

$$\begin{aligned} f(x_1), \dots, f(x_n) &\sim \mathcal{N}(0, \mathbf{K}) \\ \mathbf{K}_{ij} &= k(x_i, x_j) \end{aligned}$$

In words, the sampled values have a multivariate Gaussian distribution with a covariance matrix given by the covariance function k . This definition should be reminiscent of that of the IPP (ref.): here too we define a probability distribution over infinite-dimensional objects by constraining every finite-dimensional marginal to have the same form.

A shorthand notation is to write that

$$f \sim \mathcal{GP}(0, k)$$

and this is how we define our prior $p(f)$.

Covariance functions are often chosen to be Gaussian in shape¹⁰ (sometimes called the “squared exponential” covariance function, to avoid giving Gauss overly much credit):

$$k(x, x') = \nu \exp(-\lambda(x - x')^2)$$

It is important to be aware of the roles of the hyperparameters, here ν and λ . Since $k(x, x) = \text{Var}(f(x))$, we see that ν controls the marginal variance of f . This gives the prior a scale: for example, if $\nu = 1$, the variance of $f(x)$ is 1 for all x , and because $f(x)$ is normally distributed this implies that we do not expect f to take values much larger than 3 in magnitude. λ plays the important role of controlling how fast we expect $f(x)$ to vary: the greater λ is, the faster the covariance decays. What this implies is for very low values of λ we expect f to be locally almost constant, for very large values we expect it to vary much faster (Figure 19a). In practice it is often better (when possible) not to set the hyperparameters to pre-specified values, but infer them also from the data (see Rasmussen and Williams, 2005 for details).

One of the concrete difficulties with working with Gaussian Processes is related to the need to invert large covariance matrices when performing inference. Inverting a large, dense matrix is an expensive operation, and a lot of research has gone into finding ways of avoiding that step. One of the most promising is to approximate the Gaussian Process such that the *inverse* covariance matrix (the precision matrix) is sparse, which leads to large computational savings. Gauss-Markov processes are a class of distributions with sparse inverse covariance matrices, and the reader may consult Rue and Held (2005) for an introduction.

4.3 Details on Inhomogeneous Poisson Processes

We give below some details on the likelihood function for IPPs, as well as the techniques we used for performing Bayesian inference.

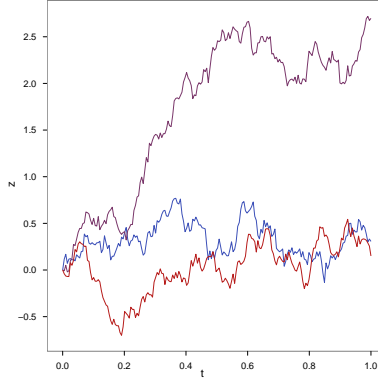
4.3.1 The likelihood function of an IPP

An IPP is formally characterised as follows: given a spatial domain Ω , e.g here $\Omega = [0, 1]^2$, and an intensity function $\lambda : \Omega \rightarrow \mathbb{R}^+$ then an IPP is a probability distribution over finite subsets S of Ω such that, for all sets $\mathcal{D} \in \Omega$,

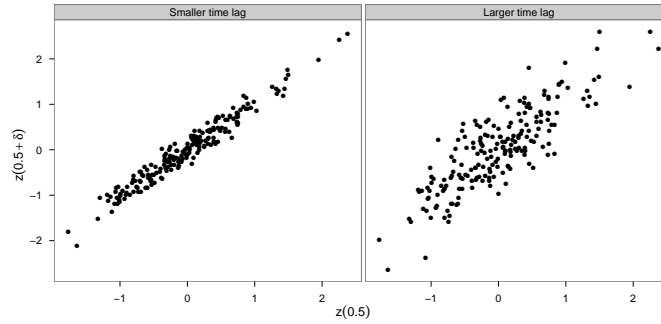
$$|S \cap \mathcal{D}| \sim \text{Poi}\left(\int_{\mathcal{D}} \lambda(x) dx\right) \quad (7)$$

⁹A GP also needs a mean function, but here we will assume that the mean is uniformly 0. See [] for details.

¹⁰For computational reasons we favour here the (also very common) Matern class of covariance functions, which leads to functions that are less smooth than with a squared exponential covariance.



(a) Stochastic processes can be used to generate random functions: here we show three realisations from a Wiener process. The Wiener process is a continuous analogue of the random walk. Although usually presented as representing the movement of a particle, one can think of the path taken by the Wiener process as a function $y(t)$, and therefore of the Wiener process as generating a probability distribution over functions. The Wiener process is a GP, but GPs used in practice generate much smoother functions (see Figure 19 below).



(b) We generated 200 realisations of the Wiener process, and plot their value at time $t = 0.5$ against their value after either a small time lag ($\delta = 0.02$), or a larger time lag ($\delta = 0.2$). The smaller the time lag, the more these values are correlated. In general, this property is reflected in the *covariance function* of the GP.

Figure 17: The Wiener Process, a member of the family of Gaussian Processes.

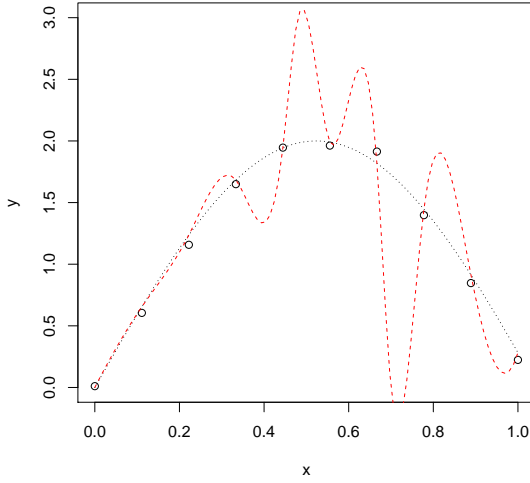


Figure 18: A non-parametric regression problem. We have measured some output y for 10 different values of some covariate x . We plot these data as open circles. Our assumption is that $y = f(x) + \epsilon$, where ϵ is zero-mean noise. We wish to infer the underlying function f from the data without assuming a parametric form for f . The two functions shown as dotted curves both have the same likelihood - they are equally close to the data. In most cases the function in red will be a far worse guess than the one in black. We need to inject that knowledge into our inference, and this can be done by imposing a prior on possible latent functions f . This can be done using a GP.

$|S \cap \mathcal{D}|$ is short-hand for the cardinal of $S \cap \mathcal{D}$, the number of points sampled from the process that fall in region \mathcal{D} . Note that in IPP, for disjoint subsets $\mathcal{D}_1, \dots, \mathcal{D}_r$ the distributions of $|S \cap \mathcal{D}_1|, \dots, |S \cap \mathcal{D}_r|$ are independent¹¹.

For purposes of Bayesian inference, we need to be able to compute the likelihood, which is the probability of the sampled point set S viewed as a function of the intensity function $\lambda(\cdot)$. If one does not mind rather extreme levels of hand-waving, the likelihood function can be derived as follows.

We note first that the likelihood can be approximated by *gridding* the data: we divide Ω into a discrete set of regions $\Omega_1, \dots, \Omega_r$, and count how many points in S fell in each of these regions. The likelihood function for the gridded data is given directly by Equation 7 along with the independence assumption: noting k_1, \dots, k_r the bin counts we have

$$p(k_1, \dots, k_r | \lambda) = \prod_{j=1 \dots r} \frac{(\lambda_j)^{k_j}}{k_j!} \exp(-\lambda_j) \quad (8)$$

$$\lambda_j = \int_{\Omega_j} \lambda(x) dx$$

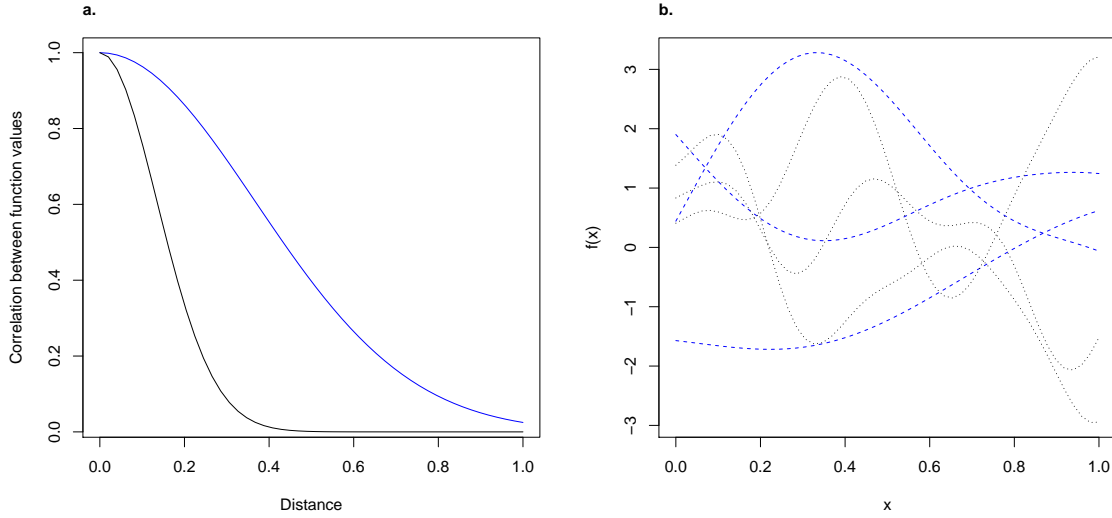
Also, since $\Omega_1, \dots, \Omega_r$ is a partition of Ω , $\prod \exp(-\lambda_j) = \exp(-\sum \lambda_j) = \exp(-\int_{\Omega} \lambda(x) dx)$.

Now comes the hand-waving argument: as we make the grid finer and finer we should recover the true likelihood, because ultimately if the grid is fine enough for all instance and purposes we will have the true locations. As we increase the number of grid points r , and the area of each region Ω_j shrinks, two things will happen:

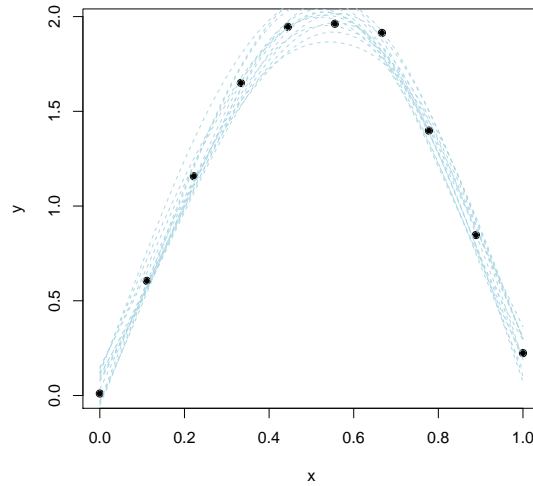
- the counts will be either 0 (in the vast majority of empty regions), or 1 (around the locations s_1, \dots, s_n of the points in S).
- the integrals $\int_{\Omega_j} \lambda(x) dx$ will tend to $\lambda(x_j) dx$, with x_j any point in region Ω_j . In dimension 1, this corresponds to saying that the area under a curve can be approximated by height x length for small intervals.

Injecting this into Equation (8), in the limit we have:

¹¹In other words, knowing how many fixations there were on the upper half of the screen should not tell you anything about how many there were in the lower half. This might be violated in practice but is not a central assumption for our purposes.



(a) GPs can be specified through their *covariance function* $k(x, x')$. The covariance function expresses the following: if we were to measure f at x and x' , how similar would we expect $f(x)$ and $f(x')$ to be? The classical *Gaussian* or *Matern* families of covariance functions impose that expected similarity go down with the distance between x and x' . **a.** Two Gaussian covariance functions with different length-scales: correlation drops faster for one than the other (shorter length-scale). **b.** Samples from the corresponding GPs: we see that a shorter length-scale leads to less smooth functions.



(b) Bayesian update of a Gaussian Process. We start with a prior distribution $p(f)$ over possible functions, then update the prior with data \mathbf{y} , to get a posterior $p(f|\mathbf{y}) \propto p(\mathbf{y}|f)p(f)$. The posterior distribution is also a probability distribution, but relative to the prior it is concentrated over the functions that are likely given the data. On this figure we show the data from Figure 18, along with functions sampled from the posterior distribution.

Figure 19: Gaussian Processes in the context of non-parametric regression. IPPs are distributions over point sets, GPs are distributions over functions. They can be used to specify a preference for “reasonable” functions.

$$p(S|\lambda(\cdot)) = \frac{1}{n} \left\{ \prod_{i=1}^n \lambda(s_i) dx \right\} \exp \left(- \int_{\Omega} \lambda(x) dx \right) \quad (9)$$

Since the factors dx and n^{-1} are independent of λ we can neglect them in the likelihood function.

4.3.2 Conditioning on the number of datapoints, computing predictive distributions

The Poisson process has a remarkable property (Illian et al., 2008): conditional on sampling n points from a Poisson process with intensity function $\lambda(x, y)$, these n points are distributed independently with density

$$\bar{\lambda}(x, y) = \frac{\lambda(x, y)}{\int \lambda(x, y) dx dy}$$

Intuitively speaking, this means that if you know the point process produced 1 point, then this point is more likely to be where intensity is high.

This property is the direct analogue of its better known discrete variant: if z_1, z_2, \dots, z_n are independently Poisson distributed with mean $\lambda_1, \dots, \lambda_n$, then their joint distribution conditional on their sum $\sum z_i$ is multinomial with probabilities $\pi_i = \frac{\lambda_i}{\sum \lambda_j}$. Indeed, the continuous case can be seen as the limit case of the discrete case.

We bring up this point because it has an important consequence for prediction. If the task is to predict where the 100 next points are going to be, then the relevant predictive distribution is:

$$p(S|S \text{ has size } n) = \prod_{i=1}^n \frac{\lambda(x_i, y_i)}{\int \lambda(x, y) dx dy} \quad (10)$$

where S is a point set of size n , whose points have x coordinates x_1, \dots, x_n and y coordinates x_1, \dots, x_n . Equation 10 is the right density to use when evaluating the predictive abilities of the model with n known (for example if one wants to compute the predictive deviance).

In the main text we had models of the form:

$$\log \lambda_i(x, y) = \eta_i(x, y) = \alpha_i + \beta_i m_i(x, y)$$

and we saw that when predicting data for a new image j we do not know the values of α_j and β_j , and need to average over them. The good news is that when n is known we need not worry about the intercept α_j : all values of α_j lead to the same predictive distribution, because α_j disappears in the normalisation in Equation 10. Given a distribution $p(\beta)$ for possible slopes, the predictive distribution is given by:

$$p(S_j|S_j \text{ has size } n) = \int p(\beta_j) \prod_{i=1}^n \frac{\exp(\beta_j m_j(x_i, y_i))}{\int \exp(\beta_j m_j(x, y)) dx dy} d\beta_j$$

It is important to realise that the distribution above does *not* factorise over points, unlike (10) above. Computation requires numerical or Monte Carlo integration over β (as far as we know).

4.4 Approximations to the likelihood function

One difficulty immediately arises immediately arises when considering Equation (9): we require the integral $\int_{\Omega} \lambda(x) dx$. While not a problem when $\lambda(\cdot)$ has some convenient closed form, in the cases we are interested in $\lambda(x) = \exp(\eta(x))$, with $\eta(\cdot)$ a GP sample. The integral is therefore not analytically tractable. A more fundamental difficulty is that the posterior distribution $p(\lambda(\cdot)|S)$ is over an infinite-dimensional space of functions - how are we to represent it?

All solutions use some form of discretisation. A classical solution is to use the approximate likelihood obtained by binning the data (Equation 8), which is an ordinary Poisson count likelihood. The bin intensities $\lambda_j = \int_{\Omega_j} \lambda(x) dx$ are approximated by assuming that bin area is small relative to the variation in $\lambda(\cdot)$, so that:

$$\lambda_j = \lambda(x_j) |\Omega_j|$$

with $|\Omega_j|$ the area of bin Ω_j and x_j . The approximate likelihood then only depends on the value of $\lambda(\cdot)$ at bin centres, so that we can now represent the posterior as the finite-dimensional distribution $p(\lambda_1 \dots \lambda_r | S)$. In practice we target rather $p(\eta_1 \dots \eta_r | S)$, for which the prior distribution is given by (see

$$\eta(x_1), \dots, \eta(x_r) \sim \mathcal{N}(0, \mathbf{K}_\theta)$$

Here \mathbf{K}_θ is the covariance matrix corresponding to the covariance function $k_\theta(\cdot, \cdot)$, and θ represents hyperparameters (e.g., marginal variance and length-scale of the process).

A disadvantage of the binning approach is that fine gridding in 2D requires many, many bins, which means that good spatial resolution requires dealing with very large covariance (or precision) matrices, slowing down inference.

Another solution, due to Berman and Turner (1992), uses again the values of $\eta(\cdot)$ sampled at r grid points, but approximates directly the original likelihood (9). The troublesome integral $\int_\Omega \lambda(x) dx$ is dealt with using simple numerical quadrature:

$$\int_\Omega \lambda(x) dx \approx \sum w_j \exp(\eta(x_j))$$

where the w_j 's are quadrature weights. The values $\lambda(s_i)$ at the sampled points are interpolated from the known values at the grid points:

$$\lambda(s_i) = \exp\left(\sum_{j=1 \dots r} a_{ij} \eta(x_j)\right)$$

the a_{ij} are interpolation weights. Injecting into (9) we have the approximate log-likelihood function:

$$\mathcal{L}(\eta) = \sum_{i,j} a_{ij} \eta(x_j) - \sum w_j \eta(x_j) \quad (11)$$

This log-likelihood function is compatible with the INLA framework for inference in latent Gaussian models (see Rue et al., 2009 and the website www.r-inla.org).

References

- Baddeley, A. and Turner, R. (2005). Spatstat: an R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42. ISSN 1548-7660.
- Baddeley, A., Turner, R., Møller, J., and Hazelton, M. (2005). Residual analysis for spatial point processes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(5):617–666.
- Berman, M. and Turner, T. R. (1992). Approximating point process likelihoods with GLIM. *Applied Statistics*, 41(1):31–38.
- Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition.
- Bruce, N. D. and Tsotsos, J. K. (2009). Saliency, attention, and visual search: an information theoretic approach. *Journal of vision*, 9(3).
- Ciuffreda, K. J. and Tannen, B. (1995). *Eye movement basics for the clinician*. Mosby, St Louis.
- Dayan, P. and Abbott, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 1st edition.
- Diggle, P. J. (2002). *Statistical Analysis of Spatial Point Patterns*. Hodder Education Publishers, 2 edition.
- Engbert, R. and Mergenthaler, K. (2006). Microsaccades are triggered by low retinal image slip. *Proceedings of the National Academy of Sciences*, 103(18):7192–7197.
- Engbert, R., Nuthmann, A., Richter, E. M., and Kliegl, R. (2005). SWIFT: A dynamical model of saccade generation during reading. *Psychological Review*, 112(4):777–813.
- Fecteau, J. H. and Munoz, D. P. (2006). Saliency, relevance, and firing: a priority map for target selection. *Trends in cognitive sciences*, 10(8):382–390.

- Gelman, A. and Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.
- Haran, M. and Tierney, L. (2012). On automating markov chain monte carlo for a class of spatial models.
- Hastie, T., Tibshirani, R., and Friedman, J. H. (2003). *The Elements of Statistical Learning*. Springer, corrected edition.
- Illian, J., Møller, J., and Waagepetersen, R. (2009). Hierarchical spatial point process analysis for a plant community with high biodiversity. *Environmental and Ecological Statistics*, 16(3):389–405.
- Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008). *Statistical Analysis and Modelling of Spatial Point Patterns (Statistics in Practice)*. Wiley-Interscience, 1 edition.
- Itti, L. (2004). Automatic foveation for video compression using a neurobiological model of visual attention. *Image Processing, IEEE Transactions on*, 13(10):1304–1318.
- Itti, L. and Koch, C. (2001). Computational modelling of visual attention. *Nature reviews. Neuroscience*, 2(3):194–203.
- Kienzle, W., Franz, M. O., Schölkopf, B., and Wichmann, F. A. (2009). Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of vision*, 9(5).
- Kliegl, R., Nuthmann, A., and Engbert, R. (2006). Tracking the mind during reading: the influence of past, present, and future words on fixation durations. *Journal of experimental psychology. General*, 135(1):12–35.
- Kruschke, J. K. (2010). *Doing Bayesian Data Analysis: A Tutorial with R and BUGS*. Academic Press, 1 edition.
- Lewis, P. A. W. and Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics*, 26(3):403–413.
- Mergenthaler, K. and Engbert, R. (2010). Microsaccades are different from saccades in scene perception. *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale*, 203(4):753–757.
- Nuthmann, A. and Henderson, J. M. (2010). Object-based attentional selection in scene viewing. *Journal of vision*, 10(8).
- Nuthmann, A., Smith, T. J., Engbert, R., and Henderson, J. M. (2010). CRISP: A computational model of fixation durations in scene viewing. *Psychological Review*, 117(2):382–405.
- Rajashekar, U., van der Linde, I., Bovik, A. C., and Cormack, L. K. (2007). Foveated analysis of image features at fixations. *Vision research*, 47(25):3160–3172.
- Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press.
- Reinagel, P. and Zador, A. M. (1999). Natural scene statistics at the centre of gaze. *Network (Bristol, England)*, 10(4):341–350.
- Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 1 edition.
- Rue, H., Martino, S., and Chopin, N. (2009). Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392.
- Schütz, A. C., Braun, D. I., and Gegenfurtner, K. R. (2011). Eye movements and perception: A selective review. *Journal of vision*, 11(5).
- Sparks, D. L. (2002). The brainstem control of saccadic eye movements. *Nature reviews. Neuroscience*, 3(12):952–964.
- Tatler, B. and Vincent, B. (2009). The prominence of behavioural biases in eye guidance. *Visual Cognition*, 17(6):1029–1054.
- Tatler, B. W. (2007). The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):1–17.

- Tatler, B. W., Hayhoe, M. M., Land, M. F., and Ballard, D. H. (2011). Eye guidance in natural vision: Reinterpreting salience. *Journal of Vision*, 11(5).
- Torrallba, A., Oliva, A., Castelhana, M. S., and Henderson, J. M. (2006). Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766–786.
- Van Der Linde, I., Rajashekar, U., Bovik, A. C., and Cormack, L. K. (2009). DOVES: a database of visual eye movements. *Spatial vision*, 22(2):161–177.
- Walther, D. and Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*, 19(9):1395–1407.
- Zelinsky, G. J. (2008). A theory of eye movements during target acquisition. *Psychological review*, 115(4):787–835.
- Zhao, Q. and Koch, C. (2011). Learning a saliency map using fixated locations in natural scenes. *Journal of vision*, 11(3).